AMENDMENTS TO THE SPECIFICATION

I. Please replace the two paragraph beginning at page 10, line 5, which starts with "In an embodiment…" with the following two amended paragraphs.

In an embodiment of the present invention, the scheduler ~~76~~ 80 as part of the external memory interface 44, is used for scheduling and issuing grants of bandwidth to network switch ports for access to the SSRAM 36. In this particular embodiment, the SSRAM 36 preferably has a 64-bit wide data path with a 100 Mb/s clock equaling a maximum 6.4 Gigabit throughput (minus a few cycles for read-write turnaround). The present implementation of 12 - 100Mb/s full-duplex ports plus a full-duplex Gigabit port 24 and an approximately 1.2 Gb/s Expansion port 30 exceeds the bandwidth capacity for accesses to the SSRAM 36. Hence, the present embodiment serves to assign portions of the bandwidth, referred to as memory access slots, to the network switch ports according to programmed information stored in an external memory device, such as an EEPROM or a controller, external to the network switch and written into an assignment table memory to be used by the scheduler ~~76~~ 80 in assigning memory access slots.

As shown in Figure 4, the network switch 12 includes an assignment table memory 100 within the external memory interface 44. The assignment table memory 100 stores an assignment table comprised of a number of programmable information entries. The assignment table is used by the scheduler ~~76~~ 80 to assign memory access slots to the ports of the network switch 12. Each programmable information entry in the assignment table contains an assignment correlating a memory access slot to either an identified port or a group of ports from which one port is selected (i.e., a slot-to-port assignment). The assignment table memory 100 may be implemented using, for example, a RAM or a group of registers.

II. Please replace the five paragraphs beginning at page 11, line 1, which starts with "As an example…" with the following five amended paragraphs.

As an example, the first configuration 202 is enlarged in Figure 5 to illustrate that each programmable information entry (e.g., 210) of the "N" number of memory access slots in the table includes a slot number 230, a port assignment 232, and a port operation code 234. Thus, as shown in Figure 5, the number "1" is stored as the slot number 230 for entry 210 and the port assignment 232 for entry 210, indicating slot one is assigned to port number 1. The second entry 212 assigns slot number 2 to the Expansion port (i.e., "E") , the third entry 214 assigns slot number 3 to port number 3, the fourth entry 216 assigns slot number 4 to the Gigabit port (i.e., "G"), and so on. Additionally, a port operation code 234 such as a read ("R") or write ("W") bit is stored in each entry of the configuration table 202 to direct the scheduler ~~76~~ <u>80</u> to assign a read or write slot, respectively, to a particular port. In other words, the port operation code 234 specifies whether the port is in a read or write operation.

Figure 5 also illustrates the port assignment 232 for a given slot may specify a port based on a detected condition. For example, entry 218 for slot number 7 in configuration 202 may be assigned to either the Expansion port 30 ("E") or the Gigabit port 24 ("G"), the "OR" condition indicated an asterisk (*). A condition, such as whether the Expansion port 30 requests a slot within a prescribed period of time, could be used to determine whether the scheduler ~~76~~ <u>80</u> assigns the slot to the Expansion port 30 or the Gigabit port 24. As a further example, the entry 220 for slot 9 in configuration 202 illustrates that a slot may be alternatively allocated to one of three (or more) ports given multiple alternative conditions. As an illustration of this example, a condition, such as which of the Expansion port 30 and the Gigabit port 24 first request the slot 9 within a first period of time, could be used to determine among these two ports which will

receive the slot. Further, if neither the Gigabit port 24 nor the Expansion port 30 request the slot within the predetermined period, slot 9 would then be assigned to port 9.

One of the configuration tables (e.g., 202) stored in the programmable storage device 102 is selected and written into the assignment table memory 100 located within the external memory interface 44 by the Host CPU 32. In an exemplary embodiment, the host CPU 32 writes a selected assignment table into the assignment table memory 100 via the PCI bus 104. After the configuration table is written into the assignment table memory 100, the scheduler ~~76~~ 80 then accesses this information via internal bus 106 and assigns the memory access slots to the ports based on the accessed information.

Figure 6 exemplifies the scheduling sequence cycle of assignment table 202 where the scheduler ~~76~~ 80 assigns memory access slots 310 in response to decoding the entries of the configuration 202 stored in the assignment table memory 100. Each slot 310 specifies the port to which to the particular slot is assigned. As indicated by arrow 302, the sequence proceeds ~~counter-clockwise~~ clockwise from slot to slot.

Additionally, a wrap-around bit ("WR") is stored in the last or "Nth" table entry of each of the configuration tables (e.g., 202, 204, 206, 208) to set the scheduling sequence cycle length. The scheduler ~~76~~ 80 will return to the first slot 310_1 in a sequence in response to detecting this wrap-around bit. For example, the configuration table entry 222 for slot N contains a wrap-around bit (WR) in the operation code entry 224 as shown in Figure 5. Figure 6 illustrates by arrow 304 that the scheduler ~~76~~ 80 returns to slot 310_1 from slot 310_N to repeat the sequence in response to the presence of the wrap-around bit (WR) stored in the configuration table entry 222. Hence, the wrap-around bit sets the "N" number of slots per cycle to a prescribed number and affords flexibility in the scheduling sequence cycle length.